Pergamon

0005-1098(94)00162-6

Automatica, Vol. 31, No. 6, pp. 903–905, 1995
Elsevier Science Ltd
Printed in Great Britain
0005-1098/95 $9.50 + 0.00

Brief Paper

Fast Subspace-based System Identification: An Instrumental Variable Approach*†

YOUNG MAN CHO‡ and THOMAS KAILATH‡

Key Words—Subspace system identification; instrumental variable; displacement structure; Schur algorithm; Lanczos algorithm.

Abstract—Recently developed subspace-based system identification (4SID) techniques have opened new routes to the identification of multi-input multi-output systems. The 4SID techniques guarantee convergence, and run faster than the statistically efficient prediction error methods without much performance loss. The resulting computational load of the 4SID techniques is $O(NM^2)$, where N is the data length and M is the sliding window size. However, the computational burden $O(NM^2)$ can become prohibitively large as N and M grow large. Noting that the major bottleneck comes from the QR factorization of an $M \times N$ data matrix and that the existing 4SID techniques do not exploit the structure of the matrices arising in the identification procedure, we propose a new implementation of the existing 4SID, which reduces the computational burden to $O(NM)$ by exploiting the displacement and low-rank structure of the matrices.

1. *Introduction*

Subspace-based state-space system identification (4SID) techniques have recently attracted much attention owing to their advantages over traditional techniques such as the prediction error (PE), output error (OE) and the instrumental variable (IV) methods (see e.g. Ljung, 1987). Of these traditional techniques, the PE method is statistically efficient. However, it has problems of convergence and long running time as the number of parameters to identify becomes large. Therefore one might expect that the PE method generally works well for single-input single-output (SISO) systems but involves various problems for multi-input multi-output (MIMO) systems. On the other hand, the 4SID techniques do not entail such difficulties as the number of parameters increases, which often makes them more suitable for MIMO system identification than the statistically efficient PE method. In many cases, the 4SID techniques perform as well as the PE method and run even faster than them as the number of parameters increases (Viberg *et al.*, 1993; Overschee and De Moor, 1994).

Among the several 4SID algorithms that have been proposed (Larimore, 1990; Verhaegan and De Wilde, 1992; Viberg *et al.*, 1993; Overschee and De Moor, 1994), this paper deals with the algorithm of Viberg *et al.* (1993), which gives a different interpretation of the algorithm of Overschee and Moor (1994). The algorithm of Viberg *et al.* can handle process noise in addition to measurement noise, and is also numerically stable. The only problem is the computational burden. Even though it runs faster than the PE method as the number of parameters increases, it requires $O(NM^2)$ flops and $O(NM)$ storage space, where N is the data length and M is the so-called sliding window size. In this paper, we propose an orders-of-magnitude faster implementation of the 4SID technique of Viberg *et al.* that only requires $O(NM)$ flops and $O(N + M^2)$ storage space. These time and storage savings are achieved by taking advantage of the Hankel (or Toeplitz) structure of the data matrix, and of the low-rank structure of the covariance matrix. Similar ideas have been used by Cho *et al.* (1994a, b). This paper shows the wide applicability of the ideas to general 4SID techniques.

Section 2 introduces the 4SID technique of Viberg *et al.* (1993). Section 3 describes a 'fast' version of that algorithm.

2. *The subspace system identification method*

A state-space model for discrete-time lumped systems is considered here:

$$\begin{aligned} x_{k+1} &= \mathbf{A}x_k + \mathbf{B}u_k + w_k \\ y_k &= \mathbf{C}x_k + \mathbf{D}u_k + v_k \end{aligned}, \qquad E\left\{\begin{bmatrix} w_i \\ v_i \end{bmatrix}\begin{bmatrix} w_j \\ v_j \end{bmatrix}^{\mathrm{T}}\right\} = \begin{bmatrix} \mathbf{Q} & \mathbf{S} \\ \mathbf{S}^{\mathrm{T}} & \mathbf{R} \end{bmatrix}\delta_{i,j}, \tag{1}$$

where x_k ($\in \mathbb{R}^{n\times 1}$) is the state of the linear system at time k, u_k ($\in \mathbb{R}^{n_i\times 1}$) is the observable input to the linear system, y_k ($\in \mathbb{R}^{n_o\times 1}$) is the observed output, w_k ($\in \mathbb{R}^{n\times 1}$) is the unobservable disturbance and v_k ($\in \mathbb{R}^{n_o\times 1}$) is the additive measurement noise. w_k and v_k are assumed to be stationary, ergodic white random processes with zero mean. $\delta_{i,j}$ denotes the Kronecker delta. The objective is to estimate the system matrices **A**, **B**, **C** and **D** from input/output data sequences $\{u_k\}$ and $\{y_k\}$ respectively; however, since the states are not directly observed, the system matrices are identifiable only to within an arbitrary non-singular state transformation. The input is assumed to be persistently exciting (Ljung, 1987). For conciseness, the notation of Viberg *et al.* (1993) is adopted and used in this paper.

The basic equation of 3SID is obtained from (1) in a matrix form

$$\mathbf{Y} = \mathbf{\Gamma}_M\mathbf{X} + \mathbf{\Phi}_M\mathbf{U} + \mathbf{N} \tag{2}$$

where M ($>n$) is a user-specified integer. $\mathbf{\Gamma}_M$ denotes the extended observability matrix, and is first estimated from the data in all existing 4SID techniques. **B** and **D** can be extracted after estimation of $\mathbf{\Gamma}_M$ in several different ways. In this paper, we confine our discussion to estimation of $\mathbf{\Gamma}_M$ and **A**.

* Received 10 August 1993; revised 14 March 1994; received in final form 22 September 1994. This paper was not presented at any IFAC meeting. This paper was recommended for publication in revised form by Associate Editor Bo Wahlberg under the direction of Editor Torsten Söderström. Corresponding author Dr Young Man Cho. Tel. +1 203 727 7255; Fax +1 203 727 7911.

† This paper is submitted for publication with the understanding that the U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of the Advanced Research Projects Agency or the U.S. Government.

‡ Information Systems Laboratory, Stanford University, Stanford, CA 94305-4055, U.S.A.

Splitting the system input and output into the past and future parts gives the equation for the future output

$$\mathbf{Y}^{\mathrm{f}} = \mathbf{\Gamma}_{\gamma}\mathbf{X}^{\mathrm{f}} + \mathbf{\Phi}_{\gamma}\mathbf{U}^{\mathrm{f}} + \mathbf{N}^{\mathrm{f}}, \tag{3}$$

and the number of block rows in $\mathbf{Y}^{\mathrm{p}}$ is denoted by β and the number of block rows in $\mathbf{Y}^{\mathrm{f}}$ by γ $(=M-\beta>n)$.

Postmultiplying (3) by an oblique projection $\mathbf{Z}^{\dagger}\mathbf{J}\mathbf{Z}$ yields

$$\mathbf{Y}^{\mathrm{f}}\mathbf{Z}^{\dagger}\mathbf{J}\mathbf{Z} = \mathbf{\Gamma}_{\gamma}\mathbf{X}^{\mathrm{f}}\mathbf{Z}^{\dagger}\mathbf{J}\mathbf{Z} + \mathbf{N}^{\mathrm{f}}\mathbf{Z}^{\dagger}\mathbf{J}\mathbf{Z}, \tag{4}$$

in which

$$\mathbf{Z} = \begin{bmatrix} \mathbf{U}^{\mathrm{p}} \\ \mathbf{U}^{\mathrm{f}} \\ \mathbf{Y}^{\mathrm{p}} \end{bmatrix}, \quad \mathbf{J} = \begin{bmatrix} \mathbf{I}_{\beta n_{\mathrm{i}}} & \mathbf{O}_{\beta n_{\mathrm{i}} \times \gamma n_{\mathrm{i}}} & \mathbf{O}_{\beta n_{\mathrm{i}} \times \beta n_{\mathrm{o}}} \\ \mathbf{O}_{\gamma n_{\mathrm{i}} \times \beta n_{\mathrm{i}}} & \mathbf{O}_{\gamma n_{\mathrm{i}} \times \gamma n_{\mathrm{i}}} & \mathbf{O}_{\gamma n_{\mathrm{i}} \times \beta n_{\mathrm{o}}} \\ \mathbf{O}_{\beta n_{\mathrm{o}} \times \beta n_{\mathrm{i}}} & \mathbf{O}_{\beta n_{\mathrm{o}} \times \gamma n_{\mathrm{i}}} & \mathbf{I}_{\beta n_{\mathrm{o}}} \end{bmatrix}, \tag{5}$$

$$\mathbf{Z}\mathbf{Z}^{\dagger}\mathbf{J}\mathbf{Z} = \begin{bmatrix} \mathbf{U}^{\mathrm{p}} \\ \mathbf{O} \\ \mathbf{Y}^{\mathrm{p}} \end{bmatrix}.$$

Since $\mathrm{span}(\mathbf{\Gamma}_{\gamma})$ is of interest, $\mathbf{Y}^{\mathrm{f}}\mathbf{Z}^{\dagger}\mathbf{J}(\mathbf{Z}\mathbf{Z}^{\mathrm{T}})^{1/2}$ can be computed instead of the unwiedly quantity $\mathbf{Y}^{\mathrm{f}}\mathbf{Z}^{\dagger}\mathbf{J}\mathbf{Z}$. We partition the singular-value decomposition (SVD) as

$$\mathbf{Y}^{\mathrm{f}}\mathbf{Z}^{\dagger}\mathbf{J}(\mathbf{Z}\mathbf{Z}^{\mathrm{T}})^{1/2} = \hat{\mathbf{P}}_{\mathrm{s}}\hat{\mathbf{S}}_{\mathrm{s}}\hat{\mathbf{V}}_{\mathrm{s}} + \hat{\mathbf{P}}_{\mathrm{n}}\hat{\mathbf{S}}_{\mathrm{n}}\hat{\mathbf{V}}_{\mathrm{n}}, \tag{6}$$

where $\hat{\mathbf{S}}_{\mathrm{s}}$ is a diagonal matrix containing the n principal singular values, and the columns of $\hat{\mathbf{P}}_{\mathrm{s}}$ are the corresponding left singular vectors. We can take $\hat{\mathbf{P}}_{\mathrm{s}}$ as $\hat{\mathbf{\Gamma}}_{\gamma}$, since $\hat{\mathbf{P}}_{\mathrm{s}} \rightarrow \mathbf{\Gamma}_{\gamma}\mathbf{T}$ as $N \rightarrow \infty$ for some full-rank matrix $\mathbf{T}$.

There are many ways of trying to find $\mathbf{A}$ and $\mathbf{C}$ from $\hat{\mathbf{P}}_{\mathrm{s}}$. Following Roy (Roy, 1987; Roy and Kailath, 1989), we shall use a total least-squares (TLS) solution. The computational costs are $O(\gamma n^2 + n^3)$ flops and $O(\gamma n)$ storage space.

3. *Fast implementation*

3.1. *Motivation.* The identification technique proposed by Viberg *et al.* has been shown to perform quite well (Viberg *et al.*, 1993; Overschee and De Moor, 1994). Although their algorithm runs faster than the existing PE methods, the computational burden of their algorithm is prohibitive for large N and M. In a typical scenario of system identification, N (the data sample size) can easily exceed 10 000. The major computational burden of the algorithm comes from computation of the quantity $\mathbf{Y}^{\mathrm{f}}\mathbf{Z}^{\dagger}\mathbf{J}(\mathbf{Z}\mathbf{Z}^{\mathrm{T}})^{1/2}$ and its SVD. To compute $\mathbf{Y}^{\mathrm{f}}\mathbf{Z}^{\dagger}\mathbf{J}(\mathbf{Z}\mathbf{Z}^{\mathrm{T}})^{1/2}$, Viberg *et al.* propose to perform one QR factorization, one matrix inversion, three matrix products and one indexing procedure ($\mathbf{J}$). Excluding the indexing procedure, these steps require $O(NM^2 + M^3)$ flops. Once $\mathbf{Y}^{\mathrm{f}}\mathbf{Z}^{\dagger}\mathbf{J}(\mathbf{Z}\mathbf{Z}^{\mathrm{T}})^{1/2}$ has been computed, it is necessary to find its SVD in order to find span $(\mathbf{\Gamma}_{\gamma})$. Here we have only to perform an economy-size SVD requiring $O(\gamma^2 M)$ flops (Golub and Van Loan, 1989; MathWorks, Inc., 1990). Overall, the computational burden of the algorithm of Viberg *et al.* becomes $O(NM^2 + M^3)$ flops.

However, these numbers are obtained without using the structure of the underlying matrices. Taking a similar approach to that used in our earlier work (Cho *et al.* 1994a, b), we propose an orders-of-magnitude faster implementation requiring only $O(NM)$ flops. In the new method, use of the generalized Schur algorithm to take advantage of the displacement structure of the underlying matrices results in fast computation of $\mathbf{Y}^{\mathrm{f}}\mathbf{Z}^{\dagger}\mathbf{J}(\mathbf{Z}\mathbf{Z}^{\mathrm{T}})^{1/2}$. The Lanczos algorithm (bi-Lanczos) efficiently determines the singular vectors corresponding to the n largest singular values by utilizing the low-rank structure of $\mathbf{Y}^{\mathrm{f}}\mathbf{Z}^{\dagger}\mathbf{J}(\mathbf{Z}\mathbf{Z}^{\mathrm{T}})^{1/2}$. Since these two steps are the major computational bottleneck in the algorithm of Viberg *et al.*, only they are discussed in the following.

3.2. *Fast computation of the covariance matrix.* In this subsection, we show how to compute the covariance matrix $\mathbf{Y}^{\mathrm{f}}\mathbf{Z}^{\dagger}\mathbf{J}(\mathbf{Z}\mathbf{Z}^{\mathrm{T}})^{1/2}$ using the fast generalized Schur algorithm, which utilizes the displacement structure of the covariance matrix. For a review of the generalized Schur algorithm and the theory of displacement structure, see Appendix A of Cho *et al.* (1994b). If $\mathbf{Z}$ has full row rank (which is the case in practice) then

$$\mathbf{Y}^{\mathrm{f}}\mathbf{Z}^{\dagger}\mathbf{J}(\mathbf{Z}\mathbf{Z}^{\mathrm{T}})^{1/2} = \mathbf{Y}^{\mathrm{f}}\mathbf{Z}^{\mathrm{T}}(\mathbf{Z}\mathbf{Z}^{\mathrm{T}})^{-1}\mathbf{J}(\mathbf{Z}\mathbf{Z}^{\mathrm{T}})^{1/2}. \tag{7}$$

We shall compute $\mathbf{Y}^{\mathrm{f}}\mathbf{Z}^{\dagger}\mathbf{J}(\mathbf{Z}\mathbf{Z}^{\mathrm{T}})^{1/2}$ as a product of $\mathbf{J}$ and triangular factors of two matrix products: $\mathbf{Y}^{\mathrm{f}}\mathbf{Z}^{\mathrm{T}}(\mathbf{Z}\mathbf{Z}^{\mathrm{T}})^{-1}$ and $(\mathbf{Z}\mathbf{Z}^{\mathrm{T}})^{1/2}$. As will be shown later, in fast estimation of $\mathbf{\Gamma}_{\gamma}$, it is not necessary to compute $\mathbf{Y}^{\mathrm{f}}\mathbf{Z}^{\mathrm{T}}(\mathbf{Z}\mathbf{Z}^{\mathrm{T}})^{-1}\mathbf{J}(\mathbf{Z}\mathbf{Z}^{\mathrm{T}})^{1/2}$ directly. Instead, some implicit form like a product of triangular factors is also acceptable.

A nice feature of our implementation is that we compute the triangular factors of $\mathbf{Y}^{\mathrm{f}}\mathbf{Z}^{\mathrm{T}}(\mathbf{Z}\mathbf{Z}^{\mathrm{T}})^{-1}$ and $(\mathbf{Z}\mathbf{Z}^{\mathrm{T}})^{1/2}$ at the same time. We form an embedded matrix whose Schur complement of the (1, 1) block becomes $\mathbf{Y}^{\mathrm{f}}\mathbf{Z}^{\mathrm{T}}(\mathbf{Z}\mathbf{Z}^{\mathrm{T}})^{-1}$, and the triangular factor of the (1, 1) block becomes $(\mathbf{Z}\mathbf{Z}^{\mathrm{T}})^{1/2}$:

$$\mathbf{W} = \begin{bmatrix} \mathbf{Z}\mathbf{Z}^{\mathrm{T}} & \mathbf{I} \\ \mathbf{Y}^{\mathrm{f}}\mathbf{Z}^{\mathrm{T}} & \mathbf{O} \end{bmatrix} = \begin{bmatrix} \mathbf{U}^{\mathrm{p}}\mathbf{U}^{\mathrm{pT}} & \mathbf{U}^{\mathrm{p}}\mathbf{U}^{\mathrm{fT}} & \mathbf{U}^{\mathrm{p}}\mathbf{Y}^{\mathrm{pT}} & \mathbf{I} & \mathbf{O} & \mathbf{O} \\ \mathbf{U}^{\mathrm{f}}\mathbf{U}^{\mathrm{pT}} & \mathbf{U}^{\mathrm{f}}\mathbf{U}^{\mathrm{fT}} & \mathbf{U}^{\mathrm{f}}\mathbf{Y}^{\mathrm{pT}} & \mathbf{O} & \mathbf{I} & \mathbf{O} \\ \mathbf{Y}^{\mathrm{p}}\mathbf{U}^{\mathrm{pT}} & \mathbf{Y}^{\mathrm{p}}\mathbf{U}^{\mathrm{fT}} & \mathbf{Y}^{\mathrm{p}}\mathbf{Y}^{\mathrm{pT}} & \mathbf{O} & \mathbf{O} & \mathbf{I} \\ \mathbf{Y}^{\mathrm{f}}\mathbf{U}^{\mathrm{pT}} & \mathbf{Y}^{\mathrm{f}}\mathbf{U}^{\mathrm{fT}} & \mathbf{Y}^{\mathrm{f}}\mathbf{Y}^{\mathrm{pT}} & \mathbf{O} & \mathbf{O} & \mathbf{O} \end{bmatrix}. \tag{8}$$

Then we partition the matrices $\mathbf{U}^{\mathrm{p}}$ into four blocks;

$$\mathbf{U}^{\mathrm{p}} = \begin{bmatrix} \mathbf{U}_0^{\mathrm{p}} & \mathbf{U}_1^{\mathrm{pT}} \\ \mathbf{U}_2^{\mathrm{p}} & \mathbf{U}_3^{\mathrm{p}} \end{bmatrix}, \tag{9}$$

where

$$\mathbf{U}_0^{\mathrm{p}} \in \mathbb{R}^{n_{\mathrm{i}} \times 1}, \quad \mathbf{U}_1^{\mathrm{p}} \in \mathbb{R}^{(N-1) \times n_{\mathrm{i}}}, \quad \mathbf{U}_2^{\mathrm{p}} \in \mathbb{R}^{n_{\mathrm{i}}(\beta-1) \times 1},$$
$$\mathbf{U}_3^{\mathrm{p}} \in \mathbb{R}^{n_{\mathrm{i}}(\beta-1) \times (N-1)}.$$

We partition $\mathbf{U}^{\mathrm{f}}$, $\mathbf{Y}^{\mathrm{p}}$ and $\mathbf{Y}^{\mathrm{f}}$ similarly. We define the displacement operator $\{\mathbf{F}_{\mathrm{f}}, \mathbf{F}_{\mathrm{b}}\}$ as follows:

$$\mathbf{F}_{\mathrm{f}} = \mathrm{diag}\,([\mathscr{S}^{n_{\mathrm{i}}} \quad \mathscr{S}^{n_{\mathrm{i}}} \quad \mathscr{S}^{n_{\mathrm{o}}} \quad \mathscr{S}^{n_{\mathrm{o}}}]),$$
$$\mathbf{F}_{\mathrm{b}} = \mathrm{diag}\,([\mathscr{S}^{n_{\mathrm{i}}} \quad \mathscr{S}^{n_{\mathrm{i}}} \quad \mathscr{S}^{n_{\mathrm{o}}} \quad \mathscr{S}^{n_{\mathrm{i}}} \quad \mathscr{S}^{n_{\mathrm{i}}} \quad \mathscr{S}^{n_{\mathrm{o}}}]),$$

where $\mathscr{S}$ is the shift-down matrix with 1s along the first subdiagonal and 0s elsewhere. Then we compute the generator of $\mathbf{W}$ with respect to $\{\mathbf{F}_{\mathrm{f}}, \mathbf{F}_{\mathrm{b}}\}$:

$$\nabla\mathbf{W}_{F_{\mathrm{f}},F_{\mathrm{b}}} = \mathbf{W} - \mathbf{F}_{\mathrm{f}}\mathbf{W}\mathbf{F}_{\mathrm{b}}^{\mathrm{T}} = \zeta_1\eta_1^{\mathrm{T}} - \zeta_2\eta_2^{\mathrm{T}} + \mathbf{R}, \tag{10}$$

where

$$\zeta_1 = [\mathbf{U}_0^{\mathrm{pT}} \quad \mathbf{U}_2^{\mathrm{pT}} \quad \mathbf{U}_0^{\mathrm{fT}} \quad \mathbf{U}_2^{\mathrm{fT}} \quad \mathbf{Y}_0^{\mathrm{pT}} \quad \mathbf{Y}_2^{\mathrm{pT}} \quad \mathbf{Y}_0^{\mathrm{fT}} \quad \mathbf{Y}_2^{\mathrm{fT}}]^{\mathrm{T}},$$
$$\eta_1 = [\mathbf{U}_0^{\mathrm{pT}} \quad \mathbf{U}_2^{\mathrm{pT}} \quad \mathbf{U}_0^{\mathrm{fT}} \quad \mathbf{U}_2^{\mathrm{fT}} \quad \mathbf{Y}_0^{\mathrm{pT}} \quad \mathbf{Y}_2^{\mathrm{pT}} \quad \mathbf{0} \quad \mathbf{0}]^{\mathrm{T}},$$
$$\zeta_2 = [(\mathscr{S}^{n_{\mathrm{i}}}\mathbf{U}^{\mathrm{p}}\mathbf{e}_N)^{\mathrm{T}} \quad (\mathscr{S}^{n_{\mathrm{i}}}\mathbf{U}^{\mathrm{f}}\mathbf{e}_N)^{\mathrm{T}} \quad (\mathscr{S}^{n_{\mathrm{o}}}\mathbf{Y}^{\mathrm{p}}\mathbf{e}_N)^{\mathrm{T}} \quad (\mathscr{S}^{n_{\mathrm{o}}}\mathbf{Y}^{\mathrm{f}}\mathbf{e}_N)^{\mathrm{T}}]^{\mathrm{T}},$$
$$\eta_2 = [(\mathscr{S}^{n_{\mathrm{i}}}\mathbf{U}^{\mathrm{p}}\mathbf{e}_N)^{\mathrm{T}} \quad (\mathscr{S}^{n_{\mathrm{i}}}\mathbf{U}^{\mathrm{f}}\mathbf{e}_N)^{\mathrm{T}} \quad \mathbf{0} \quad \mathbf{0}]^{\mathrm{T}},$$
$$\mathbf{e}_N = [0 \quad \ldots \quad 0 \quad 1]^{\mathrm{T}} \in \mathbb{R}^N.$$

$\mathbf{R}$ contains many terms, and is cumbersome to describe in matrix form. Using the block structure of $\mathbf{W}$ in (8), we can represent $\mathbf{R}$, since $\mathbf{R}$ assumes the same block structure as $\mathbf{W}$. The blocks that consist of $\mathbf{U}, \mathbf{Y}$ products turn out to have the same form. For example, the (1, 3) block of $\mathbf{R}$ corresponds to $\mathbf{U}^{\mathrm{p}}\mathbf{Y}^{\mathrm{pT}}$, and is equal to

$$\begin{bmatrix} \mathbf{U}_1^{\mathrm{pT}}\mathbf{Y}_1^{\mathrm{p}} & \mathbf{U}_1^{\mathrm{pT}}\mathbf{Y}_3^{\mathrm{pT}} \\ \mathbf{U}_3^{\mathrm{p}}\mathbf{Y}_1^{\mathrm{p}} & \mathbf{O} \end{bmatrix}. \tag{11}$$

All other terms can be obtained just by substituting the corresponding superscripts and symbols $(\mathbf{U}, \mathbf{Y})$. All other blocks of $\mathbf{R}$ (which corresponds to $\mathbf{I}$ and $\mathbf{O}$ in (8)) can be similarly obtained; for example, the (1, 4) block of $\mathbf{R}$ corresponding to $\mathbf{I}$ in (8) becomes

$$\begin{bmatrix} \mathbf{I}_{n_{\mathrm{i}}} & \mathbf{0} \\ \mathbf{0} & \mathbf{O} \end{bmatrix}.$$

It takes $O((n_{\mathrm{i}} + n_{\mathrm{o}})MN)$ flops to compute the basic building blocks of $\mathbf{R}$. Note that they can be computed directly from the data without explicitly forming $\mathbf{U}_3^{\mathrm{p}}$, $\mathbf{Y}_3^{\mathrm{p}}$, $\mathbf{U}_3^{\mathrm{f}}$ and $\mathbf{Y}_3^{\mathrm{f}}$ (Cho *et al.* 1994b). Once $\mathbf{R}$ is obtained, the generator of $\mathbf{W}$ can be computed algorithmically from (10) (Chun, 1989). The required computational load would be $O((n_{\mathrm{i}} + n_{\mathrm{o}})M)$ flops.

The displacement rank of $\mathbf{W}$ is $4n_i + 3n_o + 2$. The next step is to apply the generalized Schur algorithm to the generator of $\mathbf{W}$ to triangularize the embedded matrix $\mathbf{W}$. After $n_iM + n_o\beta$ iterations of the generalized Schur algorithm, we obtain the following equality:

$$\mathbf{W} = \begin{bmatrix} \mathbf{L}_1 \\ \mathbf{L}_2 \end{bmatrix} [\mathbf{L}_1^T \quad \mathbf{U}_2] + \begin{bmatrix} \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{S} \end{bmatrix}, \tag{12}$$

where $\mathbf{L}_1 \in \mathbb{R}^{(n_iM + n_o\beta) \times (n_iM + n_o\beta)}$ is a lower-triangular matrix. Comparing the (1, 1) block of (8) with that of (12) gives

$$\mathbf{L}_1 = (\mathbf{Z}\mathbf{Z}^T)^{1/2}.$$

Now we have to show that the same procedure results in $\mathbf{Y}^t\mathbf{Z}^T(\mathbf{Z}\mathbf{Z}^T)^{-1}$. After $n_iM + n_o\beta$ iterations of the generalized Schur algorithm applied to the generator of $\mathbf{W}$, we obtain a generator of $\mathbf{Y}\mathbf{Z}^T(\mathbf{Z}\mathbf{Z}^T)^{-1}$. Depending on the implementation, we could stop here and apply the bi-Lanczos procedure (described in Section 3.3) to $\mathbf{L}_1$ and the generator of $\mathbf{Y}\mathbf{Z}^T(\mathbf{Z}\mathbf{Z}^T)^{-1}$ by virtue of Lemma A.1 of Cho *et al.* (1994b). However, we choose to apply the generalized Schur algorithm $n_o\gamma$ times further with the generator of $\mathbf{Y}\mathbf{Z}^T(\mathbf{Z}\mathbf{Z}^T)^{-1}$ and obtain its triangular factor. This step takes $O(n_iMN + n_o\gamma N) = O(MN)$ flops.

3.3. *Fast estimation of the range of observability matrix.* It is well known that the Lanczos algorithm can be used to bi- or tri-diagonalize a matrix and compute its singular-value decomposition (SVD) (Golub and Van Loan, 1989). Owing to its inherent numerical problem, it has never been used to compute the SVD of a matrix. The conventional SVD routine can perform the same job at equivalent computational cost without sacrificing numerical stability. However, this is only true when one needs the full SVD. When only extreme singular vectors (and singular values) are required, as in our case, the Lanczos algorithm is shown to give a faster solution in a numerically stable way (as noted by Comon and Golub, 1990; Xu and Kailath, 1994).

Until now, we have computed two quantities: $(\mathbf{Z}\mathbf{Z}^T)^{1/2}$ and the triangular factors of $\mathbf{Y}\mathbf{Z}^T(\mathbf{Z}\mathbf{Z}^T)^{-1}$. The conventional SVD routine would require us to multiply these factors, which will take an additional $O(\gamma M^2)$ flops. This computational load may or may not be acceptable, depending on the relative sizes of M and N. If M (and γ) $\ll N$ then $O(\gamma M^2)$ is negligible compared with $O(NM)$, and the bi-Lanczos step that replaces the conventional SVD would not speed up the whole algorithm much. On the other hand, if M is comparable to N then, we can speed up the whole algorithm by an order of magnitude, since the proposed bi-Lanczos algorithm always gives a faster solution ($O(M^2)$) for estimation of extreme singular vectors and singular values than the conventional SVD routine. The computational gain of the bi-Lanczos step is achieved for two reasons: (i) it does not require an explicit expression of a matrix and can be applied to some factors of a matrix, and (ii) it can stop after a certain number of iterations to give singular vectors and singular values of interest.

In this subsection, we describe how to estimate $\mathbf{\Gamma}_\gamma$ (or, more specifically, span $(\mathbf{\Gamma}_\gamma)$) with $O(M^2)$ flops. We use the bi-Lanczos procedure described by Golub and Van Loan (1989). To summarize, the bi-Lanczos procedure bi-diagonalizes a matrix (given explicitly or implicitly as in a product of several matrices) through the formation of successive matrix–vector products. The important fact is that the singular values of the intermediate bi-diagonal matrix converge to those of the original matrix in descending order as the bi-Lanczos procedure proceeds. Since we have computed only the triangular factors, we can fully utilize the power of the bi-Lanczos procedure compared with the conventional SVD routine.

We apply the bi-Lanczos procedure to obtain the factorization of $\mathbf{Y}^t\mathbf{Z}^\dagger\mathbf{J}(\mathbf{Z}\mathbf{Z}^T)^{1/2}$ computed in Section 3.2. Since $\mathbf{Y}^t\mathbf{Z}^\dagger\mathbf{J}(\mathbf{Z}\mathbf{Z}^T)^{1/2}$ consists of three factors and one selection matrix ($\mathbf{J}$), we have to perform three matrix–vector products successively, at each iteration of the bi-Lanczos step. After $n + 2$ steps of the bi-Lanczos procedure, we stop and compute Rayleigh–Ritz pairs to estimate the signal subspace $\mathbf{\Gamma}_M$. Since all the factors are triangular, the matrix–vector products can be performed at half the cost of the ordinary matrix–vector product. Overall, the bi-Lanczos procedure will cost $O(n\gamma M)$, and $O(n\gamma M)$ is not much larger than $O(\gamma M)$ when $n \ll M$, which is typically the case in our application. Computing the Rayleigh–Ritz pairs takes $O(n^3)$ flops, which is again negligible. Once we have estimated the Rayleigh–Ritz pairs, it is straightforward to obtain span $(\mathbf{\Gamma}_\gamma)$. The computational load at this stage is also marginal.

Summary of the new implementation.

1. Compute a generator of the matrix $\mathbf{W}$ as shown in (10).
2. Apply the generalized Schur algorithm to this generator; after $n_iM + n_o\beta$ iterations, this will yield $\mathbf{L}_1 = (\mathbf{Z}\mathbf{Z}^T)^{1/2}$ as in (12). This also gives a generator of $\mathbf{Y}^t\mathbf{Z}^T(\mathbf{Z}\mathbf{Z}^T)^{-1}$.
3. Apply the generalized Schur algorithm to the generator found in the previous step; after $n_o\gamma$ iterations, we obtain the triangular factorization of $\mathbf{Y}^t\mathbf{Z}^T(\mathbf{Z}\mathbf{Z}^T)^{-1}$.
4. Apply the bi-Lanczos algorithm to the factorization of $\mathbf{Y}^t\mathbf{Z}^\dagger\mathbf{J}(\mathbf{Z}\mathbf{Z}^T)^{1/2}$ (obtained in steps 2 and 3) to estimate the observability matrix span $(\mathbf{\Gamma}_\gamma)$.

4. *Concluding remarks*

A new fast implementation for a subspace-based system identification technique (based on an instrumental variable approach) has been presented. This new approach exploits the displacement and low-rank + shift structures of the matrices that arise in the identification procedure to reduce the computational cost. The new implementation requires only $O(NM)$ flops compared with the $O(NM^2)$ flops of the existing techniques, with much less storage cost.

Acknowledgements—The authors wish to thank Drs Mats Viberg and Guanghan Xu for useful discussions about this paper. This research was supported in part by the Advanced Research Projects Agency of the Department of Defense and was monitored by the Air Force Office of Scientific Research under Grant F49620-93-1-0085 and in part by Rockwell International.

References

Cho, Y. M., G. Xu and T. Kailath (1994a). Fast identification of state-space models via exploitation of displacement structure. *IEEE Trans. Autom. Control*, **AC-39**, 2004–2017.

Cho, Y. M., G. Xu and T. Kailath (1994b). Fast recursive identification of state-space models via exploitation of displacement structure. *Automatica*, **30**, 45–59.

Chun, J. (1989). Fast array algorithms for structured matrices. PhD thesis, Stanford University.

Comon, P. and G. Golub (1990). Tracking a Few Extreme Singular Values and Vectors in Signal Processing. *Proc. IEEE*, **78**, 1327–1343.

Golub, G. H. and C. F. Van Loan (1989). *Matrix Computations*, 2nd ed. The Johns Hopkins University Press, Baltimore.

Larimore, W. E. (1990). Canonical variate analysis in identification, filtering and adaptive control. In *Proc. 29th IEEE Conf. on Decision and Control*, Honolulu, HI, pp. 596–604.

Ljung, L. (1987). *System Identification: Theory for the User*. Prentice-Hall, Englewood Cliffs, NJ.

MathWorks, Inc. (1990). *The PRO-MATLAB User's Guide*. MathWorks, Inc., Natick, MA.

Overschee, P. V. and B. De Moor (1994). N4SID: Subspace algorithms for the identification of combined deterministic–stochastic systems. *Automatica*, **30**, 75–93.

Roy, R. (1987). ESPRIT—estimation of signal parameters via rotational invariance techniques. PhD thesis, Stanford University.

Roy, R. and T. Kailath (1989). ESPRIT—estimation of signal parameters via rotational invariance techniques. *IEEE Trans. Acoust. Spech Sign. Process.*, **ASSP-37**, 984–995.

Verhaegen, M. H. and P. DeWilde (1992). Subspace model identification. Part I: the output-error state space model identification class of algorithms. *Int. J. Control*, **56**, 1187–1210.

Viberg, M., B. Ottersten, B. Wahlberg and L. Ljung (1993). Performance of subspace based state-space systems identification methods. In *Proc. 12th IFAC World Congress*, Sydney, Australia.

Xu, G. and T. Kailath (1994). Fast subspace decomposition. *IEEE Trans. Sig. Process.*, **SP-42**, 539–551.